Open Source und Sicherheitssoftware – ein Widerspruch?

Max Kleiner

kleiner kommunikation
CH-3001 Bern
max@kleiner.com

Abstract:

Die in vielen Kryptologie- und sensitiven Netzwerk Software Projekten enthaltenen Open Source Anteile lassen Bedenken aufkommen, dass ein Angreifer oder Konkurrent von den Kenntnissen der Algorithmen und Methoden profitieren könnte.

Allgemein denkt man, daß ein Mehr an Sicherheit durch das Verschleiern und Verstecken von Informationen und Mechanismen erreicht wird. Diese „Security by Obscurity“ hat aber einen entscheidenden Nachteil.

The enemy knows the system. -- Claude Shannon

"Der Feind kennt das System" das sollte der Ansatzpunkt sein, der als Leitfaden bei der Erstellung von Sicherheitskonzepten dienen soll. Sicherheit die nur auf der Geheimhaltung von Informationen beruht stellt sich sehr oft als trügerisch und ungenügend heraus. Das Verbessern und Härten von Algorithmen basiert vor allem auf Open Source Projekten wie OpenSSL oder True Crypt, die jederzeit in der Lage sind, Sicherheitspatches oder Empfehlungen nachzuliefern.

Damit Open Source Sicherheit in einem definierten Sinne zur Verfügung steht, braucht es die Fähigkeit einer Zertifizierung wie Common Criteria oder FIPS und die Möglichkeit die Sourcen und deren Standards jederzeit zu analysieren.

“A Crypto Pattern like AES is a proven Solution for a general Task“

1 Zusammenhang von Open Source und Security Tools aufzeigen

Dieses Open Source versus Closed Security wird aus eigener Erfahrung einer Sicherheitslösung belegt, indem die Stufen und das Vorgehensmdell einer Zertifizierung von sicherheitsrelevanter Software dargestellt wird.

Die durch Open Source gefundene Lösung lässt sich dann auch in andere Security Software einbringen und in einem der nächsten Projekte einsetzen. Aus Patterns wie RSA oder AES entstehen Komponenten und aus Komponenten wiederum entstehen Algorithmen.

Als *Ergänzung* bestehender Sicherheitskonzepte kann sich Verschleierung jedoch auch als wirkungsvoll z.B. gegenüber automatisierten Angriffen erweisen.

Gleichwohl werden immer wieder Verschlüsselungsalgorithmen geheim gehalten. Denn durch deren Kenntnis könnten die eventuellen Schwachstellen entdeckt werden, so dass sich erst später herausstellt, dass die Verschlüsselung nicht effektiv war. Offenheit der Quellen macht sich hier bezahlt, weil man als Prinzip keine Schwachstellen duldet und sich jederzeit einem Challenge stellen kann. Ein Beispiel ist RC4, welcher sieben Jahre lang geheim gehalten wurde, bis 1994 der Quellcode anonym veröffentlicht wurde und deren Schwachstelle ans Tageslicht kam.

„Security by Obscurity oder Open Source Security“